3D Reconstruction SLAM Algorithm for Mapping and Navigating

Supervisor: Limin Yu

SURF Students: Fengyuan Guo, Chenwei Pan, Xinwei Cai, Yue Ma, Ziyan Chen

SURF Code: SURF-2024-0308

School of Advanced Technology

SURF — Summer Undergraduate Research Fellowship

Abstract: Simultaneous Localization and Mapping (SLAM) is a pivotal technology in computer vision and robotics, enabling autonomous machines to navigate and incrementally map unknown environments. This technology simultaneously tracks the machine's location and orientation while mapping the surroundings, a dual challenge that has seen considerable evolution over the past two decades. **This project employs a 3D Gaussian Splatting (3DGS)-based SLAM algorithm to achieve dense mapping of environments. In the future, we plan to utilize handheld devices equipped with multiple sensors to perform multi-sensor fusion for data collection and scene reconstruction in indoor environments.**

INTRODUCTION

Background

Based on the initial 3D Gaussian Splatting, SplaTAM[1] Gaussian Splatting SLAM/ MonoGS[2], etc. are among the latest 3D GS based SLAM algorithms. These methods usually involve gradient descent, 3D Gaussians set, and dynamic addition and deletion of Gaussians strategy, Almost all strategies use the concept of key frames and introduce color error and depth error in loss. However, the selection strategy of key frames is slightly different. In optimizing Gaussians, a small number of key frames are selected to participate in the calculation of color error. In Gaussian Splatting SLAM, the current keyframe window and the randomly selected previous two frames are used. In SplaTAM, current frame, previous fame and some previous key frames that have high overlap with the current frame are selected.

Aim

- Distinguish the differences between different Gaussian splating methods.
- Simulate 3D Gaussian and MonoGS.
- Use handheld devices equipped with multiple sensors for multi-sensor fusion in indoor environments for data collection and scene reconstruction.
- Algorithm implementation and optimization.

METHODOLOGY

3D Gaussian splatting SLAM

Firstly, the input sparse point cloud is initialized to create a 3D Gaussian. Next, it works together with Sfm calibrated cameras to project 3D objects onto a 2D plane, allowing for fast α-blending for rendering. The core of the flow path is optimization. It adaptively creates a dense set of 3D Gaussians to optimize position, opacity, color and covariance. These optimized parameters are reprojected into the 2D plane, ultimately achieving high fidelity reproduction of the image.

- Definition of 3D Gauss:

$$G(x) = e^{-\frac{1}{2}(x)^T\Sigma^{-1}(x)}$$

- The covariance of the projection in 2D space:

$$\sum' = JW\sum W^T J^T$$

- Covariance of optimized 3D Gaussian:

$$\sum = RSS^T R^T$$

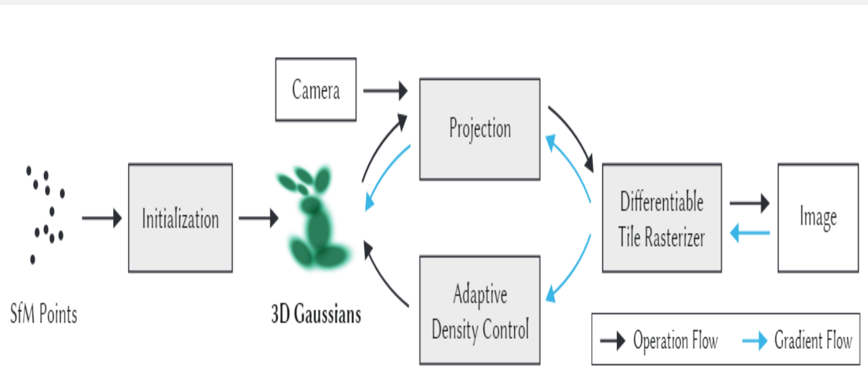


Figure.1. Optimization starts with the sparse SfM point cloud and creates a set of 3D Gaussians[3].

MonoGS

Firstly, tracking establishes and optimizes the camera pose through the images and formulas observed by the camera. Next, repeatedly select and filter keyframes in 3D reconstruction. Meanwhile, in each keyframe, a new Gaussian is inserted into the scene. Finally, mapping maintains the coherence of the 3D scene and optimizes the new Gaussian. Meanwhile, two old keyframes are randomly selected for each iteration to ensure high fidelity of the map.

- In Tracking:

The photometric residual is:

$$E_{pho} = ||I(G, T_{CW}) - \bar{I}||_1$$

The geometric residual is:

$$E_{geo} = ||D(G, T_{CW}) - \bar{D}||_1$$

The pixel depth is:

$$D_p = \sum_{i \in N} z_i \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_j)$$

- In mapping:

The isotropic regularization is:

$$E_{iso} = \sum_{i=1}^{|G|} \left\| S_i - \tilde{S_i} \cdot 1 \right\|_1$$

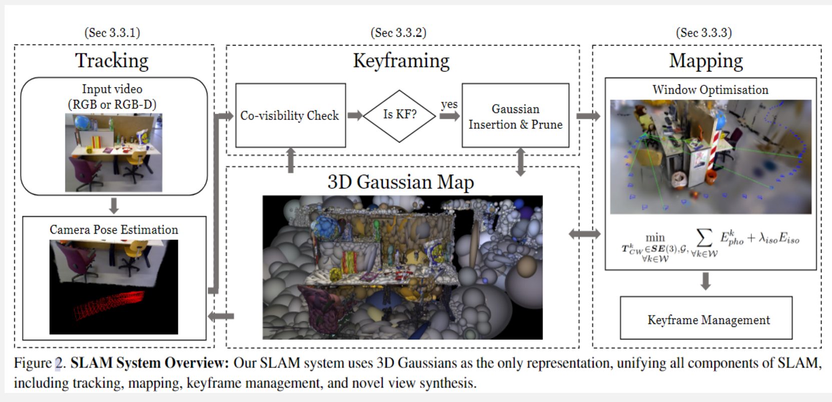


Figure.2. SLAM System Overview[2].

SIMULATION

Successful Code Reproduction and leading to generated simulation results:

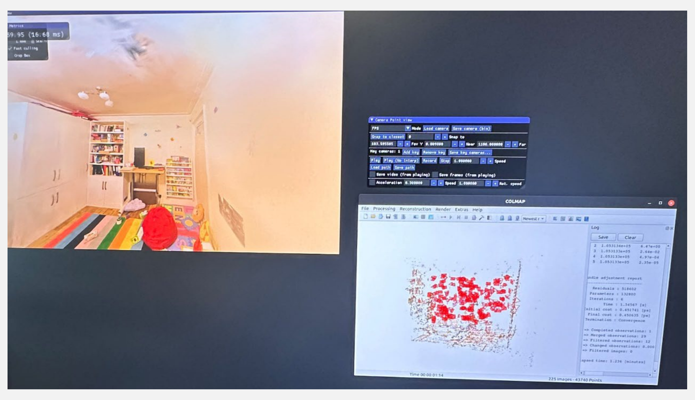


Figure.3. 3D Gaussian splatting[2].

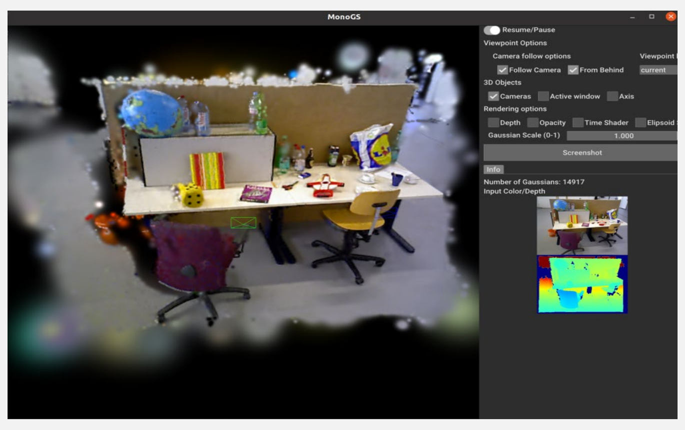


Figure.4. Tracking of MonoGS[3].

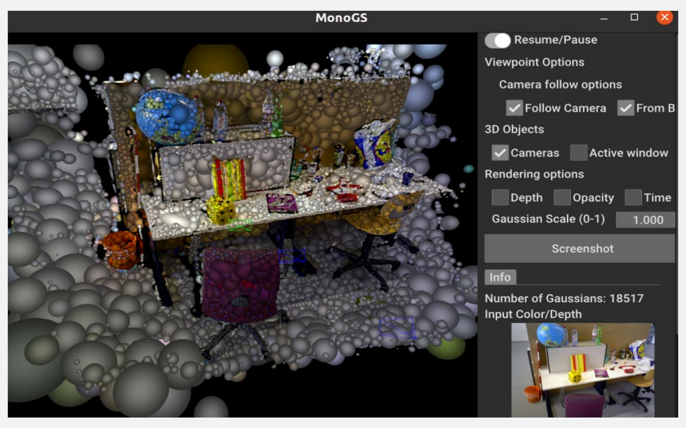


Figure.5. Keyframing of MonoGS[3].

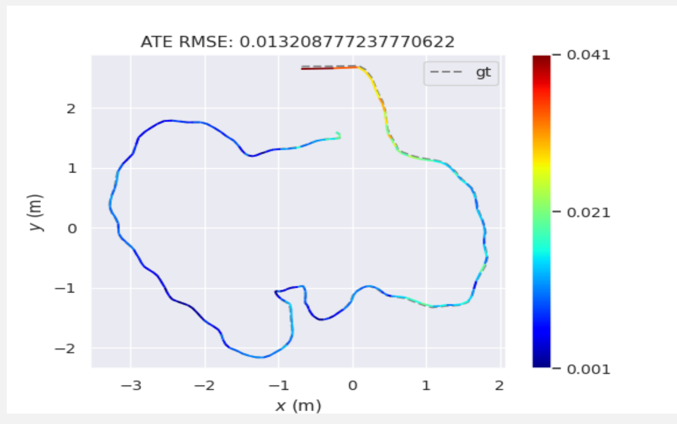
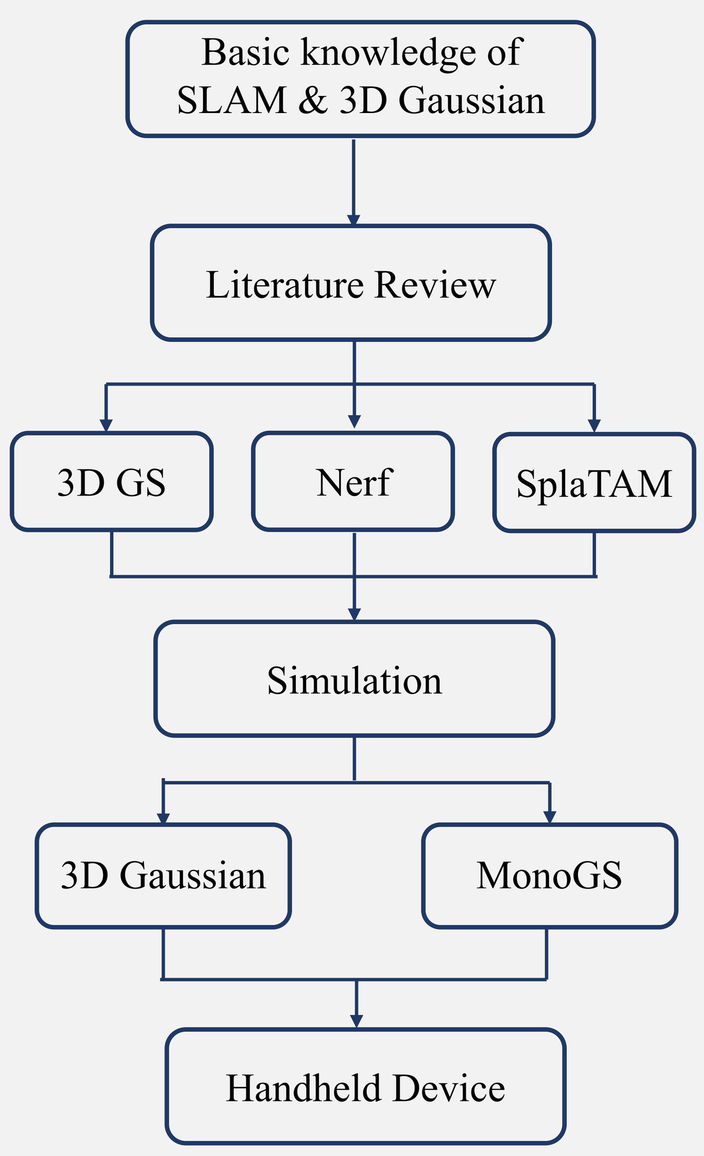


Figure.6. Simulation result of MonoGS[3].

FLOWCHART



HANDHELD DEVICE

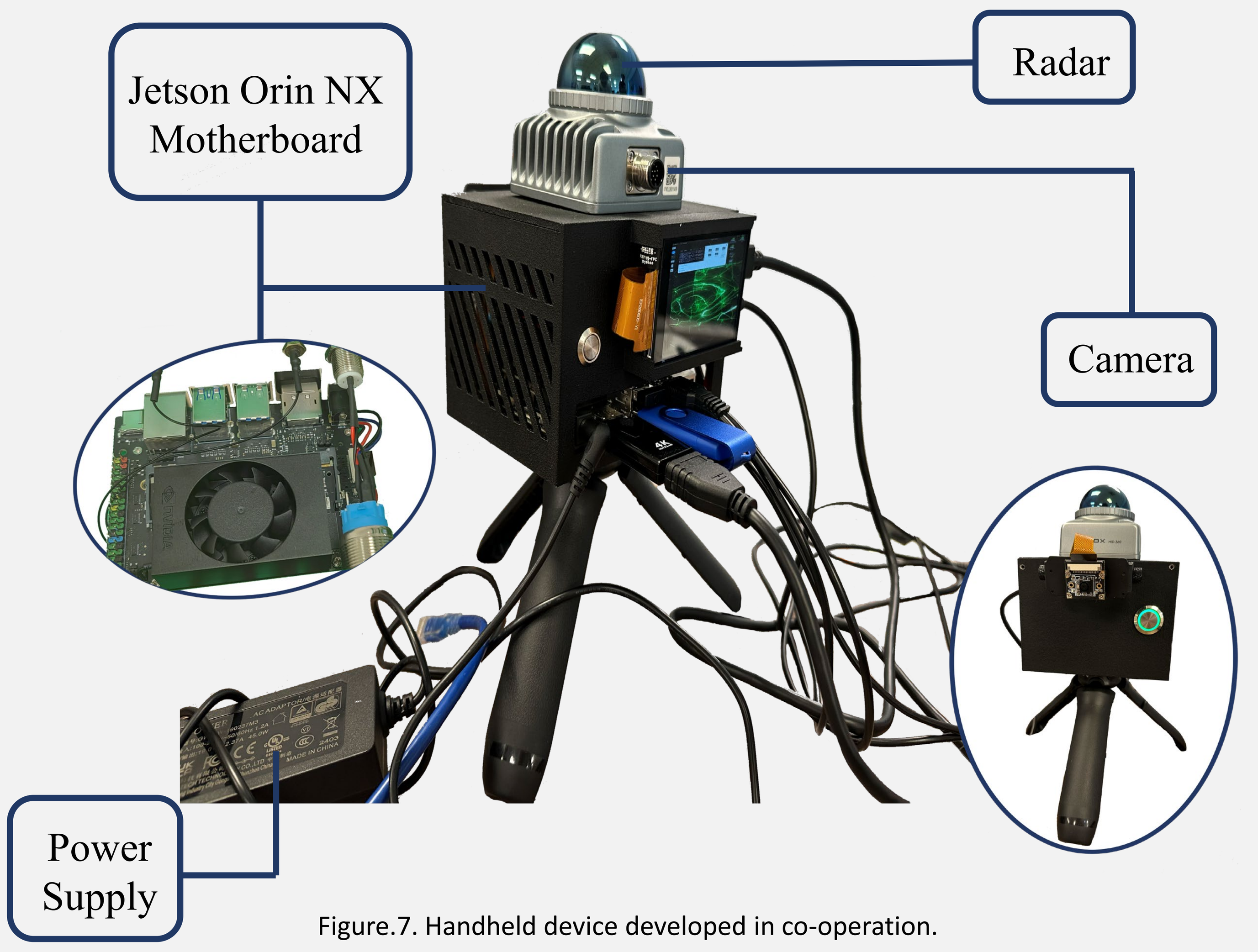


Figure.7. Handheld device developed in co-operation.

CONCLUSION

Firstly, we have completed the software installation for Ubuntu and configured the ROS, Cuda, conda, and other environments, as well as deeply learned the SLAM algorithm.

In the simulation phase, SLAM greatly improves the fidelity and diversity of captured object materials through efficient texture mapping. However, Gaussian functions cannot explicitly represent surfaces, so the closure of loops and extraction of geometric shapes in large-scale scenes will be worth future research. Gaussian splatter technology has shown outstanding performance in real-time rendering and dynamic scene representation, providing effective solutions for computer-generated images, virtual/augmented reality, robotics, film animation, retail, environmental research, and aerospace. The technology still faces some challenges, such as computational complexity, memory usage, edge artifacts, and the trade-off between real-time performance and accuracy.

Finally, our handheld device has been successfully built and tested. Its edge computing function can facilitate more convenient data collection and provide real-time output of current environment reconstruction. The integration of multiple sensors can further enhance the robustness of the 3DGS algorithm and improve the visualization of data from other sources, which is worthy of further investigation.

Acknowledgement: Special thanks to XJTLU PhD students Taoyu Wu and Lifan Xu for providing the initial codes, training and other supports during the SURF project.

References:

[1] N. Keetha *et al.* , "SplaTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM," Dec. 2023. Accessed: Aug. 1, 2024 [Online]. Available: http://arxiv.org/abs/2312.02126

[2] B. Kerbl, *et al.* , "3D Gaussian Splatting for Real-Time Radiance Field Rendering," in ACM Transactions on Graphics, vol. 42, Jul. 2023. Accessed: Aug. 1, 2024 [Online]. Available: http://arxiv.org/abs/2308.04079

[3] H. Matsuki *et al.* , "Gaussian Splatting SLAM," Dec. 2023. Accessed: Aug. 1, 2024 [Online]. Available: http://arxiv.org/abs/2312.06741

[4] C. Yan *et al.*, "GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting". 2023. Accessed: Aug. 1, 2024 [Online]. Available: http://arxiv.org/abs/2311.11700

[5] B. Mildenhall *et al*, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *Computer Vision – ECCV 2020,Lecture Notes in Computer Science*, 2020, pp. 405–421. doi: 10.1007/978-3-030-58452-8_24.